Vol.2013-CSEC-61 No.33
Vol.2013-IOT-21 No.33
2013/5/10

ストリーム暗号に適した擬似乱数生成器の提案

長尾安優美† 松尾賢志† 長瀬智行†

LFSR を用いる擬似乱生成器は安全性に関して理論的に議論がなされている．線形帰還シフトレジスタは線形システムであるため，暗号解読は比較的容易である．このような問題への対策として，LFSR のクロックを不定間隔で進める，LFSR の出力を非線形関数に組み合わせて使うなどの方法がある．今回は LFSR と非線形関数を組み合わせた擬似乱数生成器を提案し暗号論的に優れている擬似乱数を高速に生成することを示す．

A suitable pseudo-random number generator for stream cipher

AYUMI NAGAO† SATOSHI MATSUO†
TOMOYUKI NAGASE†

The linear feedback shift register method for generating pseudo-random numbers for a stream cipher is become vulnerable to cryptographic security attack. This paper introduces a new method for generating pseudo-random numbers by applying non-linear function to the output data from the LFSR to obtain sufficient security strength. To ensure the unpredictability of the generated random numbers, tests of the randomness are conducted and analyzed based on NIST randomness tool. The results show that the proposed method also offers fast generating process.

1. はじめに

暗号の一つであるストリーム暗号は，構成が比較的単純で高速処理に優れる特徴がある．近年の情報の大規模化や通信の高速化に伴い，注目を集めている暗号方式の一つである．

ストリーム暗号は平文系列と同じ長さの乱数系列を使用して逐次暗号化を行うという方式である．一般的には平文系列より短い秘密鍵と擬似乱数生成器を利用者で共有し，生成した擬似乱数系列(鍵ストリーム)を用いる． 鍵ストリームを生成する方法として，専用のアルゴリズムによるものと，ブロック暗号を元にした生成方法がある．専用アルゴリズムによる鍵ストリーム生成は線形帰還シフトレジスタ(LFSR: Linear Feedback Shift Register)に基づくものが多い．

本研究では，ストリーム暗号に使われる擬似乱数生成器について研究をしてきた．LFSR を用いる擬似乱生成器に関しては安全性に関して理論的に議論がなされている．線形帰還シフトレジスタは線形システムであるため，暗号解読は比較的容易である．このような問題への対策として，LFSR のクロックを不定間隔で進める，LFSR の出力を非線形関数に組み合わせて使うなどの方法がある．今回は LFSR と非線形関数を組み合わせた擬似乱数生成器を提案し暗号論的に優れている擬似乱数を高速に生成することを目的とする．

2. 暗号アルゴリズム強度指標

アルゴリズム固有の脆弱性がない場合，共通鍵暗号の安全性は鍵のサイズ，公開鍵暗号はベースとなる法や群のサイズによって決まる．異なる暗号アルゴリズムの強度を統一的視点で比較検討できるように導入されたのが暗号アルゴリズム強度指標である．これは，暗号アルゴリズムを最も効率のよい方法で解読した場合の解読計算量で評価する．

3. 暗号アルゴリズムの安全性

どのような共通鍵暗号アルゴリズムでも，秘密鍵の取りうる値は有限であるため，原理的には秘密鍵を 1 つずつ検査する鍵全数探索攻撃により必ず正しい鍵を見つけることができる．このため，鍵全数探索攻撃に対する耐性が共通鍵暗号アルゴリズムの安全性の上限となる．鍵全数探索攻撃に対する耐性は鍵長によって大きく左右される．

現在では，鍵全数探索に十分な耐性を持たせるため，128 ビット以上の鍵長を使うことが一般的である．既知の攻撃方法に対して，鍵を求めるために必要な計算量が鍵全数探索攻撃の計算量を下回ることがないと証明された場合に，その共通鍵暗号を安全であるとする．言い換えると，鍵全数探索よりも高速な鍵導出アルゴリズムが存在する場合，そのアルゴリズムは解読されたこととなる．

公開鍵暗号アルゴリズムの安全性は公開鍵から秘密鍵を求めることが数学的に困難であると示すことにより証明される．鍵長は安全性を左右するセキュリティパラメータである．鍵長は，数学的未解決問題が定量的に安全であると示される範囲で設定される．

† 弘前大学
Hirosaki University

4. ストリーム暗号について

ストリーム暗号は鍵ストリーム生成部と鍵ストリームと平文を結合する結合部から構成されている．秘密鍵を初期値として擬似乱数系列を生成し，平文系列との排他的論理和をとり暗号文を作る．復号の時も同じ初期値から生成した擬似乱数と暗号文との排他的論理和によって平文を得る．
ストリーム暗号は，長さが未知のデータのストリーム（電話の会話，ストリーミング・ビデオ等）の高速暗号化／復号に適している.これらにブロック暗号が使用される場合，ブロックを埋める前に終結する入力ストリームのパディングを設けなければならない.
暗号化は擬似乱数と平文の排他的論理和で構成される．ブロック暗号はブロック単位で暗号化するのでブロックサイズ分のデータがそろうまで暗号化処理ができないのに対し，ストリーム暗号は擬似乱数と平文をビット毎またはバイト毎に逐次暗号化するので待ち時間が少ない．また，常に平文サイズ=暗号文サイズとなり，データサイズが増加しないこともメリットとして挙げられる．復号時には同じ擬似乱数と暗号文との排他的論理和によって平文を得る．

5. 疑似乱数生成器について

擬似乱数生成器(Pseudo-Random Number Generator: PRNG)は固定長の初期入力から任意長のビット列を生成する確定的アルゴリズムである.
暗号技術は様々な場面での乱数を必要とする．ストリーム暗号においては擬似乱数系列
(鍵ストリーム)と平文系列の排他的論理和で構成されるため，擬似乱数生成器の強度や安全性がそのまま暗号の強さとなる．質の良い擬似乱数生成器を構築することがストリーム暗号の安全性向上の必須条件となる.
鍵ストリーム生成の方式として，線形帰還シフトレジスタ(LFSR: Linear Feedback Shift Register)を用いた方法が知られている．しかし LFSR は線形性をもつため，そのまま暗号に使用すると容易に解読されてしまう危険性がある．LFSR 型ストリーム暗号では，この出力ビット列に対して，非線形変換を施した擬似乱数列を鍵ストリームとして使用する事が一般的である.

5.1 線形帰還シフトレジスタ(LFSR)

LFSR は入力ビットが直前の状態の線形写像になっているシフトレジスタである．値を構成するビット列の一部の排他的論理和を入力ビットとする．入力に影響する出力をタップと呼ぶ.

レジスタの動作は決定的であるため，レジスタが生成する値の列はその状態によって完全に決定される．また，レジスタの取りうる状態は有限個であるため，最終的には周期的動作になる．帰還関数をうまく設定した LFSR は乱数のようなビット列を生成し，その周期も非常に長い.

5.2 暗号論的乱数とは

暗号論的乱数はストリーム暗号で用いられるほか，鍵データや初期値データなどにも用いられる．一般的な乱数が統計的性質を重視するのに対し，暗号論的乱数はさらに予測不可能性を満たさなくてはならない．これは，乱数の一部から他のビットが予測できないというものであるが，実際に予測不可能性を示すのは困難であるため，統計的乱数性や長周期性，線形複雑度という性質で代用することが多い.
ストリーム暗号は平文，暗号文に乱数列を印加することによって暗号系を構成する．乱数列に対する要求条件は，統計的なランダム性と系列の予測不可能性である．ランダム性は乱数列としては必須の条件であるが，暗号においては冗長性が高いものに対して元の情報を推定できないようにするために重要である．予測不可能性は暗号に固有の条件で，乱数列の一部を入手しても乱数列の全体を構成することは困難であることと定義される.
0,1 の値からなる系列が与えられたとき，それを生成できる LFSR の最小段数が線形複雑度である.段数が L の LFSR は 2L-1 個のデータがあればその構成は決定される.線形複雑度が大きければ大きいほど，標準回路で乱数列を生成することが困難となりストリーム暗号は安全ということになる.

5.3 SPN 構造

SPN 構造とは，換字処理(substitution)と転置処理(permutation)を繰り返し適用する構造のことである．差分解読法や線形解読法は共通暗号に対する強力な解読法である．差分解読法や線形解読法に対する安全性を示す評価指標として，最大平均差分確率と最大平均線形確率というものがある.
換字処理とは，ルールとして何種類もの置換表を用意しておき，暗号鍵のバリエーションによってどの置換表を使うかをきめる処理である．その置換表に沿ってビット列を置き換えていく.

転置処理とは，元のデータのビット列を特定のルールに基づいて並び替えるという処理である．しかし，この処理だけだと元のデータが全て 0 または 1 のとき，並び替えた結果も 0 または 1 になってしまう．そのため，転置処理は換字処理とセットで用いられるのが一般的である.

SPN 構造を応用した構造で，最後にもう一度換字処理(substitution)をする SPS 変換というものが最近の暗号で使われており，今回提案する方式にも採用している.

5.4 擬似乱数検証ツールについて

擬似乱数を検証するツールには様々なものがある．代表的なのは NIST Special Publication 800-22(NIST SP800-22)というもので，これはアメリカ国立標準技術研究所で作られた検定基準である.

Vol.2013-CSEC-61 No.33
Vol.2013-IOT-21 No.33
2013/5/10

NIST 乱数検定の手順は以下に示すように 2 つのステップにて構成される．はじめに系列毎にそれぞれ 188 の検定項目毎に p-value と呼ばれる総計量を算出する(p-value は[0,1]区間の実数値として算出される)．NIST 乱数検定では棄却率を 1%と定めているので，p-value が(1)の関係を満たせば，その検定項目(検定方式)において合格したとみなす．

$$p-value \geq 0.01$$

1 つ目のステップは個々の系列が対象であったのに対し，第 2 のステップは全系列分の集計評価が目的である．評価は 2 つの視点で実施する．

(1) UNIFORMITY(一様性)の評価

検定項目(方式)毎に p-value が等頻度に現れているか否かを判断する．具体的には p-value を等区間の 10 のクラスに分類し，各クラスの頻度が等頻度か否かについて自由度 9 の x 二乗検定を実施する．この際にも p-value(p-value2 とする)が利用される．p-value2 が(2)の関係を満たせば，検定項目は UNIFORMITY で合格したとみなす．

$$p-value2 \geq 0.0001$$

p-value は新の乱数生成器が検定対象として入力された系列よりもランダムでない系列を生成する確率と解釈できる．標準正規分布に基づいて検定が行われる場合，p-value は以下の関数 erfc を用いて計算される．

$$erfc(z) = \int_z^x \frac{2}{\sqrt{\pi}} e^{-x^2}\,dx$$

また x 二乗分布に基づいて検定が行われる場合，p-value は以下の関数 igamc を用いて計算される．

$$igamc(a,z) = \frac{1}{\Gamma(a)} \int_z^\infty e^{-t}t^{a-t}\,dt$$

$$\Gamma(a) = \int_0^\infty e^{-t}t^{a-t}\,dt$$

である．

6. 提案方式

提案方式では有限体 GF(2^8)上の演算を用いる．まず有限体 GF(2^8)の定義多項式 ϕ_8 を

$$\phi_8(x) = x^8+x^4+x^3+x^2+1 = 0x11d$$

とする．つまり GF(2^8) = GF(2)[x]/ ($\phi_8(x)$)として定義する．

有限体の元は GF(2^8)上の(すなわち{0,1}を係数に持つ 7 次以下の 1 変数多項式)として定義される．データとしては，係数を降べきに並べた 8 ビットで一つの元を表現する．つまりビット列 $b_7 \| b_6 \| b_5 \| b_4 \| b_3 \| b_2 \| b_1 \| b_0$ で多項式

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0$$

を表現する．たとえば，0x57 はビット列 0101 0111 に対応するので，多項式 $x^6+x^4+x^2+x+1$ を表す．

2 つの多項式の加算は，同じ次数ごとに係数の mod2 での加算，つまり排他的論理和として定義される．例えば，

$$\begin{aligned} 0x57 + 0xa3 &= (x^6 + x^4 + x^2 + x + 1) \\ &\quad + (x^7+x^5 + x + 1) \\ &= x^7 + x^6+x^5 + x^4+x^2 \\ &\leftrightarrow 0xf4 \end{aligned}$$

となる．

GF(2^8)の元 $f(x)=\Sigma a_i x^i$ と x との乗算 x・f(x)を

$$\sum a_i x^{i+1} \bmod \phi_8(x)$$

で定義する．例えば，

$$\begin{aligned} 0x02 \cdot 0x87 &= x \cdot (x^7+x^2 + x + 1) \\ &= x^8+x^3+x^2 + x \\ &= (x^4+x^3+x^2 + 1) + x^3 + x^2 + x \\ &= x^4 + x + 1 = 0x13 \end{aligned}$$

となる．

x^i・f(x)は上記定義から帰納的に計算できる．

GF(2^8)の任意の 2 つの元 $f(x)=\Sigma a_i x^i$,$g(x)=\Sigma b_i x^i$ の乗算 f・g は

$$f \cdot g(x) = \sum_{i=0}^{14} \sum_{j=0}^{i} (a_j\ b_{i-j}) x^i \bmod \phi_8(x),$$

で定義する．

擬似乱数生成器に関しては，その擬似乱数生成器によって得られる複数の乱数系列について検定を行い，(1)p-value が 0.01 以上になる割合と(2)p-value の一様性によって擬似乱数生成器の検定が行われる．

(1)については乱数列の個数を m としたとき, p-value が 0.01 以上になる割合が

$$0.99 \pm 3\sqrt{\frac{0.99 \times 0.01}{m}}$$

の範囲にある場合，良い擬似乱数生成器であると判定される．

(2)については，区間[0,1]を 10 分割し，各区間に属する p-value の個数が均等であるかどうかを x 二乗分布によって検定する．具体的には $1 \leqq i \leqq 10$ について，Fi を区間[(i-1)/10,i/10]

に属する p-value の個数とするとき，

$$x^2 = \sum_{i=1}^{10} \frac{(F_{i-m}/10)^2}{m/10}$$

を計算し，p-value=igamc(9/2,x^2/2)を計算する．p-value $\geqq$ 0.0001 のとき，良い擬似乱数生成器であると判定される．

6.1 提案方式のアルゴリズム

鍵長 256 ビット(128 ビット LFSR2 つ), サブ鍵 b=48 バイトの擬似乱数生成器である．提案方式は λ 関数と ρ 関数の二つから構成される．

Vol.2013-CSEC-61 No.33
Vol.2013-IOT-21 No.33
2013/5/10

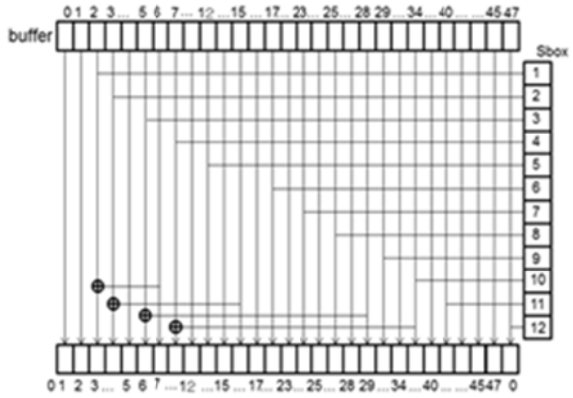

図 1　提案方式　λ関数

Figure 1　Proposed method for function (λ)

λ関数では S-box を更新させるため，1 バイトずつバッファをシフトしたり XOR の計算をしている．

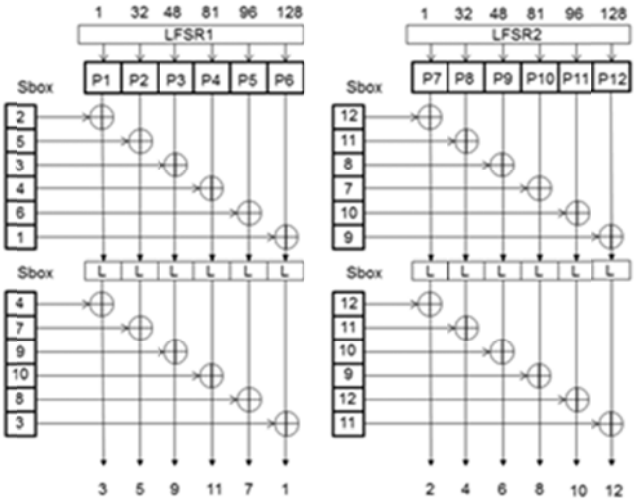

図 2　提案方式　ρ 関数

Figure 2　Proposed method for function (ρ)

LFSR1,2 ともに 1 ビットシフトさせる毎に P1~P12 に 1 ビットずつ格納する．それを 8 回繰り返し 8 ビットのデータを格納する．P1~P12 に格納する LFSR のビットを取ってくる場所を以下のように定義する．

$$P1 = (LFSR1_0), P2 = (LFSR1_{31}), P3 = (LFSR1_{47}),$$
$$P4 = (LFSR1_{79}), P5 = (LFSR1_{95}), P6 = (LFSR1_{127}),$$
$$P7 = (LFSR2_0), P8 = (LFSR2_{31}), P9 = (LFSR2_{47}),$$
$$P10 = (LFSR2_{79}), P11 = (LFSR2_{95}), P12 = (LFSR2_{127})$$

P1~P12 を S-box に通し，線形変換をしてもう一度 S-box に通す. それらを上記の図の順番に連結させて 96 ビットの鍵ストリームとする．これが提案方式における疑似乱数生成の 1 サイクルである．

線形変換 L は GF(2^8)上の 2×2 行列で，次の式で表す．

$$\begin{pmatrix} v_0 \\ v_1 \end{pmatrix} = L(u_0, u_1) = \begin{pmatrix} 1 & 1 \\ 1 & d \end{pmatrix} = \begin{pmatrix} u_0 \\ u_1 \end{pmatrix} \qquad d \in GF(2^8)$$

S-box(S8)の構造

提案方式では 8 ビット入出力の S-box(S8)を用いる．

4 ビット入出力の S-Box(S4)を以下で定義する．

$$S_4[16] = \{1,3,9,10,5,14,7,2,13,0,12,15,4,8,6,11\}$$

線形変換: l

$$l(x \quad y) = \begin{pmatrix} 1 & e \\ e & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \qquad e = 0x04$$

SPS 変換の構成の概要は上記の図に従う．つまり，

$$y_0 = S_4[S_4[x_0] \oplus e \cdot S_4[x_1] \oplus 0xa],$$
$$y_1 = S_4[e \cdot S_4[x_0] \oplus S_4[x_1] \oplus 0x5]$$

最後に全体を 1 ビット巡回シフト演算によって有限体上の処理としての対称性を崩す．

$$S_8[x] = (y_0 || y_1) <<<_8 \quad 1$$

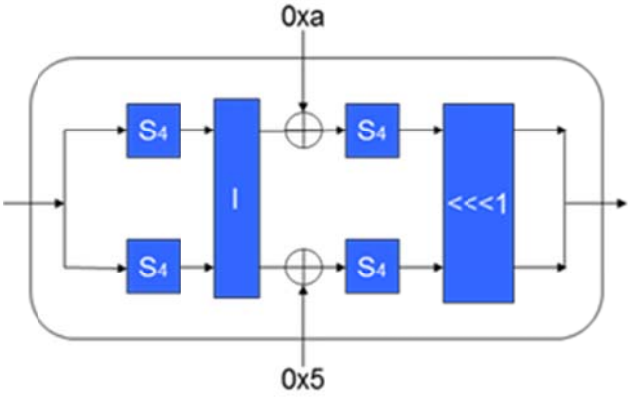

図 3　S-box(S8)の概要図

Figure 3 Overview of S-box structure

6.2 鍵の与え方

一般に，確定アルゴリズムで定義される擬似乱数生成器の特性として，出力される乱数列は秘密鍵とサブ鍵の組み合わせにより一意に決まる．故に全く同じ組み合わせのものを用いてはならない．特に同じ秘密鍵を用いて複数の鍵ストリームを生成する場合には，異なるサブ鍵を用いる必要がある．

また，鍵の値を全て 0 にしてしまうと，LFSR の性質上繰り返し 0 が出力されることになるため全て 0 という鍵は不適である．

6.3 暗号化と復号化

提案方式を用いてストリーム暗号を実現する場合，平文系列を 1 バイトずつに分割し，生成した乱数列と排他的論理和をすれば良い．復号化も同様にして実現できる．

6.4 提案方式の安全性について

鍵全数探索攻撃に対する耐性は鍵長によって大きく左右される．提案方式では，推奨値の 128 ビットより多い 256 ビットとしているので全数探索に対しては安全性があると言える．また周期についても，LFSR の既約多項式に基づいて最長となるようにタップする位置を決めているので問題

ない. S-box に関しても Enocoro と同じものを使っており，暗号の強度の指標となる最大線形確率は$2^{-4.00}$，最大差分確率は$2^{-4.678}$となる．非線形部分に関しての安全性は Enocoro と同等なものだということができる.

7. 実行結果

プログラムの実行結果として，擬似乱数を発生させる処理時間について記す．今回の実験では，120000 バイトの擬似乱数を生成するまでにかかる時間を clock(Visual C++)を用いて 100 回測定し，平均を求めた.

	提案方式	Enocoro-128v2
平均	4.07746 秒	15.54653 秒

提案方式は Enocoro の約 3.812 倍速い時間で同数の乱数が生成できるという結果が得られた.

次に提案方式によって生成された系列を NIST 乱数検定にて検定した結果を記す．今回は 10 万ビット×1000 本，100 万ビット×1000 本の 2 つの系列について，検定を行った．表の空欄は合格，「X」は不合格，「－」は系列数が推奨値に満たず検定を行わなかったことを表している.

表 1 NIST 乱数検定に含まれる検定法一覧

Table 1 A statistical package of the NIST Test Suite

1	一次元度数検定
2	ブロック単位の頻度検定
3	累積和検定
4	連の検定
5	ブロック単位の最長列検定
6	2 値行列ランク検定
7	離散フーリエ検定
8	重なりのないテンプレート検定
9	重なりのあるテンプレート検定
10	Maurer のユニバーサル検定
11	近似エントロピー検定
12	ランダム偏差検定
13	種々のランダム偏差検定
14	系列検定
15	線形複雑度検定

表 2 乱数検定の結果

Table 2 Random numbers tests results

A: 10 万ビット×1000 本　P:　合格比率　　○:　合格

B: 100 万ビット×1000 本　U:　一様性　　X:　不合格

	A		B	
	P	U	P	U
1	○	○	○	○
2	○	○	○	○
3	○	○	○	○
4	○	○	○	○
5	○	○	○	○
6	○	○	○	○
7	○	○	○	○
8	-	-	○	○
9	-	-	○	○
10	-	-	○	○
11	○	○	○	○
12	-	-	○	○
13	-	-	○	○
14	○	○	○	○
15	-	-	○	○

8. 考察

今回の評価結果において，10 万ビット×1000 本，100 万ビット×1000 本(NIST 推奨値)のいずれの評価でも全ての検定に合格した．これらの結果により，NIST SP 800-22 による検定では提案方式によって生成された乱数列の統計的な偏りは発見できなかった.

また，乱数生成の速度についてであるが，提案方式は Enocoro の約 3.812 倍速い時間で同数の乱数が生成できる．提案方式では 1 サイクルにかかる時間は LFSR を導入した分遅くなっているが, その 1 サイクルで 12 バイトずつ出力しているので実行するサイクル数が減少し結果的に速度向上につながったと考えることができる.

しかし，真のランダム性を有しているデータを NISTSP800-22 の検定に通した場合でも, 検定項目を全てが合格する確率は 54%しかなく，いずれか一つの項目のみが不合格となる確率は 33%となる．このことから NISTSP800-22 の検定だけで乱数の善し悪しを判断するのではなく，今後はストリーム暗号の様々な解読法の観点から安全性を評価していくことが必要であると考える.

参考文献

1) 廣瀬勝一, 疑似乱数生成系の検定方法に関する調査　調査報告書，京都大学，http://www.cryptrec.go.jp/estimation/rep_ID0207.pdf, (2004).

2) 東京理科大学理工学部電気電子情報工学科　金子研究室疑似乱数生成系の検定方法に関する調査報告書(2004).

3) 鴻巣慧, 武藤健一郎, 古市洋希, 渡辺大, 金子敏信, “Enocoro-128 ver.1.1 の再同期攻撃耐性評価,” IEICE Tech. rept, IT2007, No.50, pp.7-13，2008.

4) 小池慎一，山住富也，“2 次 M 系列の線形複雑度について,” 愛知工業大学研究報告第 39 号 B，pp.141-145，2004.

5) 奥富秀俊，金田学，山口健二，中村勝洋，“NIST 乱数検定を用いた乱数性能の評価について” IEICE Tech. Rept, IT2005, No.165, pp.79-84，2008.

6) 田中圭介，暗号のための基礎理論，The Operations Research Society of Japan, pp.131-136, (2009).